readme.md for @git.zone/tsview

A powerful developer tool for browsing and managing S3-compatible storage and MongoDB databases through a sleek web UI — with real-time change streaming baked in. Built with TypeScript, designed for developers who need quick, visual access to their data stores during development.

Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit community.foss.global/. This is the central community hub for all issue reporting. Developers who sign and comply with our contribution agreement and go through identification can also get a code.foss.global/ account to submit Pull Requests directly.

Install

```
# Global installation (recommended for CLI usage)
pnpm add -g @git.zone/tsview

# Local installation (for programmatic usage)
pnpm add @git.zone/tsview
```

□□ Quick Start

1. Configure Your Connection

Create a `.nogit/env.json` file in your project root (auto-generated by `gitzone service`):

```
{
  "S3_ENDPOINT": "localhost",
  "S3_PORT": "9000",
  "S3_ACCESSKEY": "minioadmin",
  "S3_SECRETKEY": "minioadmin",
  "S3_USESSL": false,
  "MONGODB_URL": "mongodb://localhost:27017",
  "MONGODB_NAME": "mydb"
}
```

2. Launch the Viewer

```
tsview
```

That's it! Your browser opens automatically to the viewer interface.

□ Features

□□ S3 Storage Browser

Powered by `dees-storage-browser` from `@design.estate/dees-catalog`:

- **Column View Navigation** — Mac Finder-style interface with resizable columns
- **List View** — Traditional key-based view with hierarchical navigation
- **Real-time Preview** — View images, JSON, text, code, and more directly in the browser
- **Bucket Management** — Create, delete, and switch between buckets
- **File Operations** — Upload, download, delete, move, and copy objects
- **In-place Text Editing** — Edit text files directly with change tracking
- **Smart Content Type Detection** — Automatic recognition for 20+ file types
- **Breadcrumb Navigation** — Clickable path traversal

□□ MongoDB Browser

- **Database Explorer** — Hierarchical navigation through databases and collections
- **Database Overview** — Collection counts, data sizes, index stats at a glance
- **Document Viewer** — Paginated table view with JSON filter support
- **Document Editor** — Full CRUD with syntax-highlighted code editor and change tracking

- **Index Management** — View, create, and drop indexes
- **Aggregation Pipeline** — Run aggregation queries directly
- **Collection Stats** — Document counts, sizes, storage metrics
- **Server Status** — Connection info, version, uptime
- **Show/Hide System Databases** — Toggle visibility of `admin`, `local`, `config`

⚡ Real-Time Change Streaming

- **MongoDB Change Streams** — Live updates via native MongoDB change streams
- **S3 Change Detection** — Polling-based bucket monitoring with ETag comparison (5s intervals)
- **Activity Stream** — Combined timeline of all changes from both sources, filterable by type
- **Live Indicators** — Green dot + change count badges on active views
- **WebSocket Subscriptions** — Per-collection, per-bucket, or global activity feed
- **Auto-Reconnect** — Subscriptions automatically restored after connection loss

□□ Modern Web UI

- Dark theme designed for developer comfort
- Responsive layout with resizable panels
- Context menus for quick actions
- Everything bundled — zero external runtime dependencies in the browser

CLI Usage

```
# Start viewer with auto-detected port (starts from 3010)
tsview

# Force a specific port
tsview --port 3000

# S3 browser only
tsview s3

# MongoDB browser only
tsview mongo
# or
```

```
tsview mongodb
```

Programmatic API

Use tsview as a library in your own tools:

```
import { TsView } from '@git.zone/tsview';

const viewer = new TsView();

// Option 1: Load from .nogit/env.json (gitzone service format)
await viewer.loadConfigFromEnv();

// Option 2: Configure programmatically
viewer.setStorageConfig({
  endpoint: 'localhost',
  port: 9000,
  accessKey: 'minioadmin',
  accessSecret: 'minioadmin',
  useSsl: false,
});

viewer.setMongoConfig({
  mongoDbUrl: 'mongodb://localhost:27017',
  mongoDbName: 'mydb',
});

// Option 3: Cloud services
viewer.setStorageConfig({
  endpoint: 's3.amazonaws.com',
  accessKey: 'AKIAXXXXXXX',
  accessSecret: 'your-secret-key',
  useSsl: true,
  region: 'us-east-1',
});

viewer.setMongoConfig({
```

```
  mongoDbUrl: 'mongodb+srv://user:pass@cluster.mongodb.net',
  mongoDbName: 'production',
});

// Start the server
const port = await viewer.start();
console.log(`Viewer running on http://localhost:${port}`);

// Or specify a port
await viewer.start(3500);

// Graceful shutdown
await viewer.stop();
```

Configuration

Project-level via `.smartconfig.json`

```
{
  "@git.zone/tsview": {
    "port": 3015,
    "killIfBusy": true,
    "openBrowser": false
  }
}
```

Option	Type	Default	Description
`port`	`number`	auto	Fixed port (auto-finds from 3010 if not set)
`killIfBusy`	`boolean`	`false`	Kill existing process if port is busy
`openBrowser`	`boolean`	`true`	Automatically open browser on start

Port priority: CLI `--port` flag → `.smartconfig.json` → auto-detect

Environment Variables (`.nogit/env.json`)

S3

Variable	Description
`S3_ENDPOINT`	S3-compatible server hostname
`S3_PORT`	Server port (optional)
`S3_ACCESSKEY`	Access key ID
`S3_SECRETKEY`	Secret access key
`S3_USESSL`	Use HTTPS (`true`/`false`)

MongoDB

Variable	Description
`MONGODB_URL`	Full connection string (preferred)
`MONGODB_NAME`	Default database name

Or use individual variables:

Variable	Description
`MONGODB_HOST`	Hostname
`MONGODB_PORT`	Port
`MONGODB_USER`	Username
`MONGODB_PASS`	Password
`MONGODB_NAME`	Database name

Supported S3 Providers

tsview works with any S3-compatible storage:

Provider	Status
MinIO	Perfect for local dev
AWS S3	Amazon's object storage
DigitalOcean Spaces	Simple object storage

Provider	Status
Backblaze B2	S3-compatible API
Cloudflare R2	Zero egress fees
Wasabi	Hot cloud storage
Self-hosted	Any S3-compatible server

Supported File Types for Preview

Category	Extensions
Images	`.png`, `.jpg`, `.jpeg`, `.gif`, `.webp`, `.svg`
Text	`.txt`, `.md`, `.log`, `.sh`, `.env`
Code	`.json`, `.js`, `.ts`, `.tsx`, `.jsx`, `.html`, `.css`
Data	`.csv`, `.xml`, `.yaml`, `.yml`
Documents	`.pdf`

□□ Architecture

```
tsview/
├── ts/                          # Backend (Node.js)
│   ├── api/                     # TypedRequest API handlers
│   │   ├── handlers.s3.ts       #   S3 bucket & object operations
│   │   └── handlers.mongodb.ts  #   MongoDB CRUD & admin operations
│   ├── config/                  # Configuration management
│   ├── server/                  # Web server (TypedServer + TypedSocket)
│   ├── streaming/               # Real-time change streaming
│   │   ├── classes.changestream-manager.ts  # MongoDB + S3 watchers
│   │   └── interfaces.streaming.ts          # Subscription interfaces
│   ├── interfaces/              # Shared TypeScript interfaces
│   └── tsview.classes.tsview.ts  # Main class
├── ts_web/                      # Frontend (bundled via esbuild → base64ts)
│   ├── elements/                # Web components (LitElement)
│   │   ├── tsview-app.ts        #   App shell + navigation
│   │   ├── tsview-mongo-*.ts    #   MongoDB browser components
│   │   └── tsview-activity-stream.ts  # Real-time activity feed
```

```
│   ├── adapters/                # Data provider adapters
│   │   └── s3-data-provider.ts  #   IStorageDataProvider for dees-storage-browser
│   ├── services/                # API + WebSocket clients
│   ├── styles/                  # Dark theme
│   └── utilities/               # Formatting helpers
└── .smartconfig.json            # Build & runtime config
```

How It Works

1. **Backend** — A `TypedServer` serves the bundled web UI and exposes a typed API via `TypedRequest` over HTTP. A `TypedSocket` WebSocket layer handles real-time streaming subscriptions.
2. **Frontend** — LitElement-based web components communicate with the backend via `TypedRequest`. The S3 browser uses `dees-storage-browser` from `@design.estate/dees-catalog` with a custom `IStorageDataProvider` adapter. The `ChangeStreamService` connects over WebSocket and distributes real-time events to active views via RxJS Subjects.
3. **Streaming** — The `ChangeStreamManager` creates MongoDB Change Streams and S3 BucketWatchers on demand (one per subscribed resource). Changes are pushed to subscribed clients and accumulated in a 1000-event ring buffer for the Activity Stream view.

Development

```
# Clone
git clone https://code.foss.global/git.zone/tsview.git
cd tsview

# Install dependencies
pnpm install

# Build (bundles frontend + compiles TypeScript)
pnpm build

# Development mode with hot reload
pnpm run watch

# Run tests
pnpm test
```

License and Legal Information

This repository contains open-source code licensed under the MIT License. A copy of the license can be found in the license file.

Please note: The MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as required for reasonable and customary use in describing the origin of the work and reproducing the content of the NOTICE file.

Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH or third parties, and are not included within the scope of the MIT license granted herein.

Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines or the guidelines of the respective third-party owners, and any usage must be approved in writing. Third-party trademarks used herein are the property of their respective owners and used only in a descriptive manner, e.g. for an implementation of an API or similar.

Company Information

Task Venture Capital GmbH Registered at District Court Bremen HRB 35230 HB, Germany

For any legal inquiries or further information, please contact us via email at hello@task.vc.

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.

Revision #4
Created 2026-03-28 11:08:52 UTC by foss.global Team
Updated 2026-03-28 12:15:34 UTC by foss.global Team